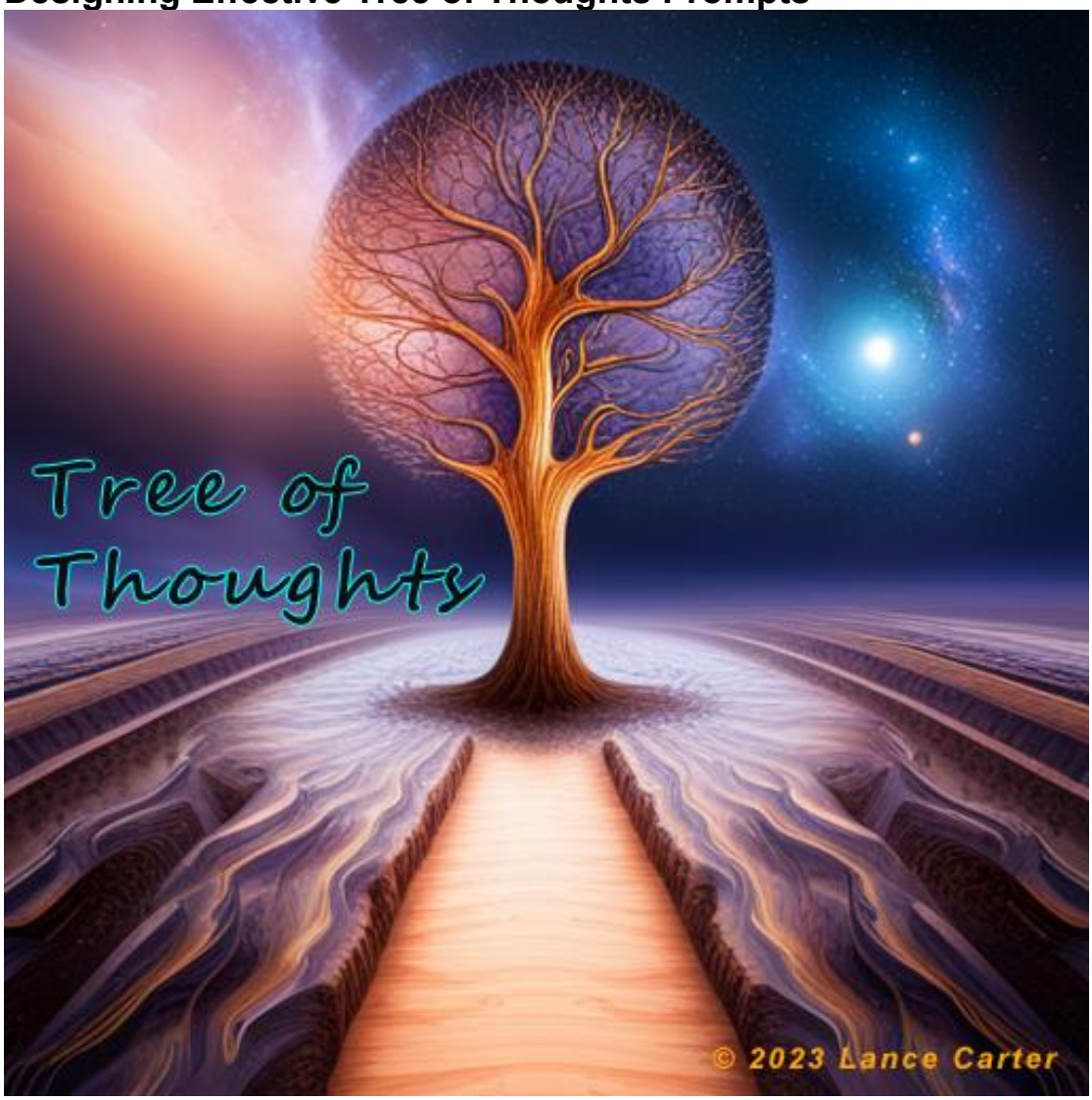
Tree of Thoughts

© 2023 Lance Carlyle Carter
(This paper is an excerpt from ‘Assembly Time’ by the author.)

Designing Effective Tree of Thoughts Prompts



Abstract:

The Tree of Thoughts (ToT) framework offers a powerful tool for complex problem-solving and interactive exploration. This is a "methodological guide" or a "practical implementation guide" that draws from a recent paper[1].
It focuses on explaining and providing instructions on how to apply the proposed methods and concepts discussed in the paper. It serves as a practical resource for readers interested in utilizing the techniques and frameworks described in the paper. While it may contain elements of a review by summarizing and discussing the content of the original paper, its primary focus is on guiding readers on how to implement and utilize the proposed methodologies effectively. We demonstrate the application of ToT prompts in the domain of scientific problem-solving, focusing on physics and chemistry.

We outline the systematic steps involved in constructing effective ToT prompts, including problem definition, determining the number of thought paths and steps, and evaluating the plausibility of each step. Furthermore, we provide a summary algorithm for creating ToT prompts, which involves breaking down complex problems, guiding the AI through sequential reasoning, and evaluating thought paths. By leveraging the ToT methodology, researchers and practitioners can effectively simulate human brainstorming processes and enhance decision-making in scientific problem-solving.

Keywords: Tree of Thoughts (ToT), Problem-solving, Interactive exploration, Complex problems, Prompt design, Thought paths, Sequential reasoning, Plausibility evaluation, Scientific problem-solving, AI modeling, Decision-making, Problem decomposition, Value assessment, Search algorithms.

Introduction:

The Tree of Thoughts (ToT) framework has emerged as a valuable approach to engage AI models in complex problem-solving tasks. While its potential applications span various domains, this paper focuses on the utilization of ToT prompts in scientific problem-solving, specifically in the fields of physics and chemistry. By deconstructing intricate problems into manageable steps and exploring multiple thought paths, ToT prompts enable a comprehensive exploration of potential solutions. This paper aims to provide a user guide for scientists and researchers seeking to leverage the ToT framework in their scientific endeavors, emphasizing the importance of problem decomposition, prompt design, value assessment, and efficient search algorithms.

SYSTEM Input Structure:

(USER: Summarize the Tree of Thought framework and then make a user guide on how to use Tree of Thought prompts in science:)

ChatGPT4: In order to utilize the Tree of Thoughts (ToT) framework to its full potential and better interact with me, consider the following steps:

****SYSTEM Input Structure****

The SYSTEM input should be structured with a clear problem and number of thoughts or steps required to solve the problem. When using ToT, tasks should be clearly defined in terms of:

1. **Problem Definition**: What is the exact task or problem to solve? For instance, when looking for an equality equation in physics, the problem could be, "Identify the possible equality equations from the given physical constants and variables."

2. **Number of candidates and thoughts/steps**: How many ways do you want me to think about this problem? For example, you could ask for three different thoughts or steps toward possible solutions.

3. **Candidate Evaluation**: How should I assess the feasibility of each thought path? ToT applies a degree of judgment such as "sure/maybe/impossible" to evaluate how likely an intermediate thought path leads to the solution.

Let's illustrate this with a sample input for examining the effect of gravity on an object in motion, "Explore three different ways to solve for the trajectory of a projectile given an initial angle of launch and velocity. In each path, include three steps of thinking and evaluate each step as 'sure', 'maybe', or 'impossible' for it to lead to the correct solution."

(USER: Create a specific example of the Tree of Thoughts framework:)

Let's take a specific example and describe it through the Tree of Thoughts framework.

Problem Definition: "Given the initial launch velocity as 15 m/s and the angle of launch as 45 degrees, calculate the trajectory of the projectile. Explore three different ways to solve this using different physical parameters."

****Method 1: Using the basic kinematic equations****

1. _Step 1:_ "Calculate the initial components of velocity. This involves breaking down the velocity into its horizontal (v_x) and vertical (v_y) components using trigonometry. The vertical component will be affected by gravity. 'Sure'—this step is a standard part of projectile motion calculations.
2. _Step 2:_ "Calculate the time of flight using the vertical component of velocity and acceleration due to gravity. 'Sure'—this step is also a standard part of calculations in basic mechanics.
3. _Step 3:_ "Calculate the horizontal range of the projectile using time of flight and horizontal velocity. 'Sure'—this step allows us to find the trajectory path."

****Method 2: Using the angular kinematic equations****

1. _Step 1:_ "We could also use the angle of projection to calculate the range (R) directly using the formula 'R = ($v^2 \sin 2\theta$)/g', where v is the initial velocity, θ is the angle of launch, and g is the acceleration due to gravity. 'Maybe'—while this formula is valid, it skips several intermediate steps that could provide useful context.
2. _Step 2:_ "We could calculate the maximum height the projectile reaches with the formula 'H = ($v^2 \sin^2\theta$)/(2g)'. 'Sure'—this step is necessary to complete the projectile's trajectory.
3. _Step 3:_ "Combining the range and maximum height, we can then plot the parabolic path of the projectile. 'Sure'—this step, combined with the previous calculations, allows us to find the trajectory path."

****Method 3: Using energy principle****

1. _Step 1:_ "The potential energy of the object at the highest point of the journey equals the kinetic energy it had at the beginning. 'Maybe'—this approach is less direct for calculating trajectory, but still possible.
2. _Step 2:_ "Under this principle, we can figure out the maximum height. 'Sure'—this is an accurate way to find the maximum height achieved by the projectile.
3. _Step 3:_ "We can then determine the time it takes to reach this height. 'Sure'—understanding the time it takes to reach the peak can then allow for the determination of the total flight time if we multiply it by 2, which subsequently could lead us to understanding the full trajectory."

This illustration goes through three different methods using the Tree of Thoughts model, which are fairly common in solving problems of this type. This multidirectional approach represents the true strength of the ToT framework, allowing for robust exploration of potential solutions, each evaluated independently. Further steps might then involve averaging the results or checking for any deviation in the solutions suggested by the different methods.

(USER: Talk about how to ask questions about the nature of the task and if the definition of the task fits into a logical information context or is it somewhat vague or dealing with other things. What can the USER do to make the problem definition clear and concise and easy to understand? What language do you look for or use to define problems and in what context do certain things come into play?)

ChatGPT4: Indeed, a crucial step in solving any problem, whether simple or complex, is having a clear and concise problem definition. As an AI, I need clear, precise, and structured input to produce the best possible output. Here are some methods to ensure clarity when defining the problem:

Be Specific: Try to avoid vague or ambiguous wording. For instance, rather than saying "Identify equations in physics," specify the type of equations needed. This could be something like "Identify potential energy-related equations involving mass, gravity, and height."

Ensure Completeness: Ensure that the problem includes all necessary information needed to resolve it, and eliminates irrelevant details. For instance, if you'd like to understand the potential energy of an object at a certain height, make sure to specify the object's mass and the height in question.

Ask Clarifying Questions: If you're unsure about the nature or scope of the problem, posing it as a question can provide clarity. For example, "What would be the potential energy of a 10kg object on top of a 100m high hill?" This helps to validate the problem context and eliminate possible misunderstandings.

Use Standard and Recognizable Terminology: Whenever possible, use universally accepted or standard terms associated with the problem at hand to ensure mutual understanding. For instance, if you're discussing a physics problem, use standard physical terms and units like "force," "acceleration," "m/s," and "kg." Avoid using jargon or colloquial expressions that could lead to ambiguity.

Define the Desired Output: Clearly specify what you expect the resolution or solution to look like. This gives me a targeted endpoint to work toward. For example, instead of asking for help to "solve an equation," it may be better to ask to "find the value of the variable X in the equation".

Contextualize the Problem: To ensure that I correctly interpret your problem, it may be helpful to situate it within a broader context. For example, if you give me an equation and ask me to solve it without further context, I can do that. But if you tell me that this equation arises in the context of physics, in a specific scenario, I can provide a solution that's tailored more precisely to your needs.

The language I look for to define problems involves straightforward, precise wording and specific terms related to the field at hand (like "acceleration," "equilibrium," "inertia" for physics). Problem defining also depends on the context: in a scientific research context, I would look for terms related to the specific research topic or field; in a business context, I would pay attention to industry-specific keywords or terms related to organizational goals and strategies.

Moreover, the command itself should indicate a direct request and highlight the problem-solving nature of the task. For example, "Please calculate...", "Find the solution for...", or "Identify the possible..." are clear instructions that demonstrate the need for problem-solving.

Lastly, remember that dialogue is a two-way process. If the initial problem definition is unclear, I may ask follow-up questions, or you might need to provide additional information, revise, or refine the prompt. Hence, maintaining the interaction can help us collaborate on refining and eventually solving the problem.

(USER: Talk about how the USER can evaluate the number of steps needed for the possible solution or what possible steps to get to partial solutions that help build progress. How can we evaluate how complex you need to get on first run and on subsequent problems:)

ChatGPT4: When using the Tree of Thoughts (ToT) framework, the number of candidates (possible solutions or thoughts paths) and the number of steps (intermediate stages within a thought path) that you choose essentially dictates the complexity and depth of the problem-solving process.

Number of Candidates: There's no defined maximum or minimum number of candidates you can choose. It primarily depends on the complexity of the

problem at hand and the computational resources available. The more candidates you have, the broader the search scope, and the higher the chances of finding novel or creative solutions to complex problems.

However, increasing the number of candidates also exponentially increases the computational load and the time taken for the problem-solving process. If too many candidates are chosen, it may also lead to an information overload situation where it becomes relatively harder to identify the best solution among many.

For most practical problems, choosing three to five candidates can strike a useful balance between exploratory breadth and manageable workload.

Number of Steps: The number of steps is essentially the depth of the problem-solving process. A higher number of steps would mean a more detailed, step-by-step solution, but it would also take longer to generate and evaluate. Less complex problems might require fewer steps, while complex problems may necessitate more detailed breakdowns.

(USER: What is the format of asking you to frame your answers in the 'sure/maybe/impossible' format? Define these assessments.)

The feasibility assessment in the Tree of Thoughts framework allows the user to evaluate the potential of a thought path to lead to a solution. Specifically, based on the intermediate results generated at each step, the system can assess if the current path being followed is likely (sure), potentially (maybe), or unlikely (impossible) to lead to a successful resolution of the problem.

"Sure" Evaluation: This rating should be assigned when a thought or step significantly contributes to the solution's formation or seems to be leading on the right track. Essentially, this thought or step seems quite likely to help solve the problem.

"Maybe" Evaluation: This rating signifies a thought or step that could potentially lead to a solution but isn't as clear-cut or definitive as a 'Sure' thought. These tend to be speculative or exploratory paths that require further investigation.

"Impossible" Evaluation: If a thought or step seems unlikely to lead to a solution or doesn't hold up under scrutiny, it would be categorized as 'Impossible'. This thought pathway may have a logical flaw, be based on incorrect assumptions, or simply be irrelevant to the problem at hand.

As for the format, when using Tree of Thoughts, you can instruct me to self-evaluate my thoughts. For example, you might ask, "Generate three different thought paths to solve this problem and evaluate each step as either 'sure', 'maybe', or 'impossible' in regard to its potential to contribute to a solution."

In this sense, the assessment pushes me to generate steps or thoughts for each path, and at the same time, I perform an initial evaluation, providing a

degree of confidence on how each thought contributes to the problem's resolution.

Remember, these categories ('sure', 'maybe', 'impossible') are meant to provide an initial reflexive assessment and guide the exploration process. Still, the final determination of a thought's validity would come after the entire process is complete, allowing for a rich exploration and possible backtrack or re-routing along the way.

This type of evolving and reflexive approach can be especially helpful for complex, unclear, or open-ended problems where the solution pathway is not immediately apparent, and the system might need to explore various possibilities and strategies before converging on a solution.

****Creating ToT Prompts****

Creating effective ToT prompts involves breaking down a complex problem into smaller, manageable chunks. These prompts should ideally guide the AI through multi-step reasoning, exploration, or problem-solving. Here are some example prompts:

1. "Consider three ways to derive the equality equation for energy in an ideal simple harmonic oscillator. What are the steps for each method? At each step, evaluate the plausibility of it leading to the correct solution."

2. "Imagine three physicists approach solving the same dimensional analysis problem. Each physicist writes down their steps. Share these steps and evaluate the likely success of each one."

3. "In interpreting the results of a scientific experiment, provide three different interpretations. Assess how likely each one is to be close to the true meaning of the results, writing 'sure', 'maybe', or 'impossible'."

****Summary or Algorithm for the ToT Prompts Structure****

A potential algorithmic structure for creating ToT prompts could look like this:

1. Define your complex problem.
2. Decide on the number of thought paths and steps within each.
3. Create prompts that break down the problem into these multiple paths, and guide the conversion of these paths into manageable and sequential steps.
4. Specify an evaluation criteria for each step in each thought path. This could be a rating or tagging system like 'sure', 'maybe', or 'impossible'.
5. Launch your AI processing with these ToT prompts and wait for the solutions/thoughts to be generated.
6. Collect the AI responses, analyze them according to your set evaluation criteria, and then backtrack on unfruitful thought paths if necessary to hone your final decision-making.

Remember that the power of the ToT methodology lies in its ability to simulate human brainstorming and problem-solving processes by exploring

and backtracking on a variety of thought possibilities. By carefully constructing your ToT prompts, you can more effectively utilize the AI for complex problem-solving tasks.

Tree of Thoughts User Guide

User guide on using ToT prompts in science:
Identify the scientific problem you want to solve that requires non-trivial search or planning. For example, designing a new molecule with certain properties, figuring out an experimental procedure, etc.

1. Identify the scientific problem you want to solve that requires non-trivial search or planning. For example, designing a new molecule with certain properties, figuring out an experimental procedure, etc.
2. Decompose the problem solving process into coherent "thought" steps. For a chemistry experiment, thoughts could be high-level actions like "choose solvent", "set up apparatus", "add reagent A".
3. Design a "propose prompt" that generates potential next thoughts from a given state. Prompt the LM to propose multiple options.
4. Design a "value prompt" that evaluates states. Prompt the LM to deliberate on progress made towards the end goal. Aggregate values over samples.
5. Choose a search algorithm like BFS or DFS based on the problem structure. Systematically explore the most promising thoughts.
6. Track thoughts explored over time. The chain of thoughts leading to the solution can serve as an explainable plan or procedure.
7. Try different prompt designs, search params, thought sizes to optimize performance. Quantitatively evaluate solutions if possible.
8. For open-ended problems, pick the solution with highest LM-assessed value, or have an expert qualitatively evaluate.

(USER: Describe the BFS and DFS search algorithms:)

AI: Search algorithms like Breadth-First Search (BFS) and Depth-First Search (DFS) are commonly used in problem-solving tasks, including those within the Tree of Thoughts (ToT) framework. The choice of which algorithm to use depends on the problem structure and the specific requirements of the problem at hand.

Breadth-First Search (BFS):
BFS is a systematic search algorithm that explores a problem space by expanding all possible successors of a given state before moving on to the next level. It starts at the initial state and explores all immediate neighboring states before moving to the next depth level. BFS ensures that all states at a given depth level are visited before proceeding to deeper levels. This algorithm guarantees the shortest path to a solution when applied to problems with uniform cost or no specific preference for depth.

Depth-First Search (DFS):
DFS, on the other hand, explores a problem space by going as deep as possible along each branch before backtracking. It starts at the initial state

and systematically explores each path until it reaches a leaf node or a condition for termination. DFS prioritizes depth over breadth and may traverse long paths before backtracking. It is often used when the depth of the problem space is unknown, or when the solution is more likely to be found at deeper levels.

The selection between BFS and DFS for the ToT framework can be influenced by the nature of the problem being solved. BFS is well-suited for problems that require considering multiple thought paths simultaneously, evaluating the feasibility of each step within each path. It ensures a comprehensive exploration of the problem space and can be effective when the problem structure does not have a specific depth bias.

On the other hand, DFS may be suitable when the problem structure has a natural depth bias, and exploring a single thought path in depth is more fruitful. DFS can be useful in situations where backtracking is required, allowing the algorithm to efficiently navigate through large problem spaces by focusing on a single path at a time.

In the context of the ToT framework, the choice between BFS and DFS as the search algorithm depends on the complexity of the problem, the desired exploration strategy, and the nature of the thought paths to be followed. Researchers can select the most appropriate algorithm based on these factors to ensure an effective exploration and solution discovery process within the ToT framework.

Summary and Conclusion:

The Tree of Thoughts (ToT) framework serves as a versatile tool for complex problem-solving and interactive exploration in the scientific domain. By structuring problems, generating coherent thought steps, and employing effective prompts, researchers can leverage AI models to navigate intricate scientific challenges. The systematic approach outlined in this paper highlights the significance of problem decomposition, prompt design, and thoughtful evaluation of thought paths. Through careful implementation of ToT prompts, scientists can harness the power of AI to simulate human brainstorming processes, optimize decision-making, and drive scientific progress. Future research can focus on refining the ToT methodology, exploring diverse scientific domains, and quantitatively evaluating the performance of ToT-based solutions.

REFERENCE:
Tree of Thoughts:
Deliberate Problem Solving with Large Language Models
2305.10601.pdf (arxiv.org)

Assembly Time
by Lance Carlyle Carter
https://www.amazon.com/dp/B0C6VJDSYD
ISBN-13: 978-1-935057-32-1